

# Práctica 4 SWAP

XuSheng Zheng

## Índice

<b>1. Certificado autoafirmado SSL</b>	<b>2</b>
1.1. Opciones avanzadas . . . . .	2
<b>2. Apache con certificado SSL</b>	<b>3</b>
<b>3. Nginx como balanceador para peticiones HTTPS</b>	<b>5</b>
3.1. Opciones avanzadas . . . . .	6
<b>4. IPTABLES</b>	<b>7</b>
4.1. Opciones avanzadas . . . . .	8
4.2. Hacer persistente reglas IPTABLES . . . . .	9
<b>5. Bibliografía</b>	<b>11</b>

## 1. Certificado autoafirmado SSL

Empezamos habilitando el módulo SSL de Apache y creando el directorio para ubicar los certificados:

```
kuzheng@ml-kuzheng:~$ sudo a2enmod ssl
(sudo) password for kuzheng:
considering dependency setenvif for ssl:
Module setenvif already enabled
considering dependency mime for ssl:
Module mime already enabled
considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
kuzheng@ml-kuzheng:~$ sudo service apache2 restart
kuzheng@ml-kuzheng:~$ sudo mkdir /etc/apache2/ssl
```

Ahora creamos el certificado con los siguientes datos:

```
kuzheng@ml-kuzheng:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/swap_kuzheng.key -out /etc/apache2/ssl/swap_kuzheng.crt
Generating a RSA private key
.....+++++
writing new private key to '/etc/apache2/ssl/swap_kuzheng.key'
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank.
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SWAP
Organizational Unit Name (eg, section) []:P4
Common Name (e.g. server FQDN or YOUR name) []:kuzheng
Email Address []:kuzheng@correo.ugr.es
```

### 1.1. Opciones avanzadas

De las opciones introducidas anteriormente, podemos prescindir de **nodes** para añadir una contraseña a la clave. Además, tenemos las siguientes opciones que pueden ser interesantes:

- **-config filename**: permite especificar un archivo de configuración para la creación de los certificados.
- **-subj arg**: permite especificar datos para la creación del certificado. El argumento debe ser de la forma **/type0=value0/type1=value1/type2=.....**. Al introducir esta opción no nos pedirá los datos como habíamos hecho.
- **-addext ext**: permite añadir una extensión de x509 en el certificado generado.

Como ejemplo creamos el siguiente certificado:

```
kuzheng@ml-kuzheng:~$ sudo openssl req -x509 -days 365 -newkey rsa:2048 -subj "/C=ES" -keyout /etc/apache2/ssl/swap_kuzheng2.key -out /etc/apache2/ssl/swap_kuzheng2.crt
Generating a RSA private key
.....+++++
writing new private key to '/etc/apache2/ssl/swap_kuzheng2.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
kuzheng@ml-kuzheng:~$
```

Podemos ver que nos pide la contraseña para generar el certificado. Si eliminamos la opción **-subj** podemos ver que aparte de pedirnos la contraseña nos pide también el resto de informaciones:

```
kuzheng@ml-kuzheng:~$ sudo openssl req -x509 -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/swap_kuzheng3.key -out /etc/apache2/ssl/swap_kuzheng3.crt
Generating a RSA private key
.....+++++
writing new private key to '/etc/apache2/ssl/swap_kuzheng3.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank.
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:_
```

## 2. Apache con certificado SSL

Editamos el archivo de configuración `/etc/apache2/sites-available/default-ssl.conf`:

```
# Enable/Disable SSL for this virtual host.
SSLEngine on

SSLCertificateFile /etc/apache2/ssl/swap_xuzheng.crt
SSLCertificateKeyFile /etc/apache2/ssl/swap_xuzheng.key

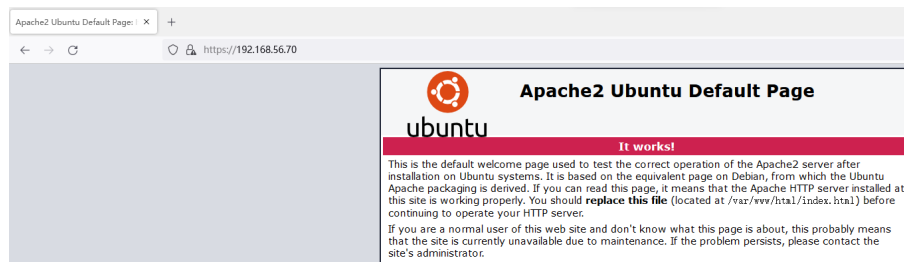
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convinience.
SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt
```

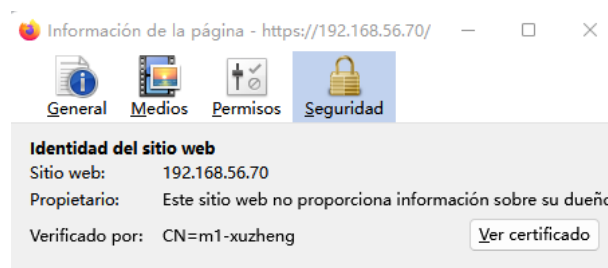
Activamos **default-ssl** y reiniciamos Apache:

```
xuzheng@m1-xuzheng:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
systemctl reload apache2
xuzheng@m1-xuzheng:~$ sudo service apache2 reload
```

Para comprobar que se ha instalado correctamente el certificado accedemos desde el navegador del anfitrión:



Accedemos al certificado desde el candado con exclamación a la izquierda de la URL:



## Certificado

xuzheng	
<b>Nombre del asunto</b>	
País	ES
Estado/Provincia	Granada
Localidad	Granada
Organización	SWAP
Unidad organizativa	P4
Nombre común	xuzheng
Dirección de correo electrónico	xuzheng@correo.ugres
<b>Nombre del emisor</b>	
País	ES
Estado/Provincia	Granada
Localidad	Granada
Organización	SWAP
Unidad organizativa	P4
Nombre común	xuzheng
Dirección de correo electrónico	xuzheng@correo.ugres
<b>Validez</b>	
No antes	Fri, 21 Apr 2023 15:56:27 GMT
No después	Sat, 20 Apr 2024 15:56:27 GMT

Ahora procedemos a copiar los archivos que hemos creado anteriormente a m2 y m3:

```
xuzheng@m1-xuzheng:~$ sudo scp -P 2222 /etc/apache2/ssl/swap_xuzheng.crt xuzheng@192.168.56.71:/home/xuzheng/swap_xuzheng.crt
The authenticity of host '192.168.56.71:2222 ([192.168.56.71]:2222)' can't be established.
ECDSA key fingerprint is SHA256:nHPleschFa+XOPJfTLtyFv0/5Vha0kPEVOVingdcQL8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.71:2222' (ECDSA) to the list of known hosts.
xuzheng@192.168.56.71's password:
swap_xuzheng.crt                                100% 1424    2.0MB/s   00:00
xuzheng@m1-xuzheng:~$ sudo scp -P 2222 /etc/apache2/ssl/swap_xuzheng.key xuzheng@192.168.56.71:/home/xuzheng/swap_xuzheng.key
xuzheng@192.168.56.71's password:
swap_xuzheng.key                                100% 1704    974.5KB/s  00:00
xuzheng@m1-xuzheng:~$ sudo scp /etc/apache2/ssl/swap_xuzheng.crt xuzheng@192.168.56.72:/home/xuzheng/swap_xuzheng.crt
The authenticity of host '192.168.56.72 (192.168.56.72)' can't be established.
ECDSA key fingerprint is SHA256:UJztKnHkQidJkezhQhI4r5F5pIUVAQX0e0iU+KSh3o.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.72' (ECDSA) to the list of known hosts.
xuzheng@192.168.56.72's password:
swap_xuzheng.crt                                100% 1424    2.2MB/s   00:00
xuzheng@m1-xuzheng:~$ sudo scp /etc/apache2/ssl/swap_xuzheng.key xuzheng@192.168.56.72:/home/xuzheng/swap_xuzheng.key
xuzheng@192.168.56.72's password:
swap_xuzheng.key                                100% 1704    2.8MB/s   00:00
xuzheng@m1-xuzheng:~$
```

Al igual que en m1, en m2 creamos el directorio `/etc/apache2/ssl`, copiamos los archivos en dicho directorio y configuramos **default-ssl**:

```
xuzheng@m2-xuzheng:~$ sudo mkdir /etc/apache2/ssl
[sudo] password for xuzheng:
xuzheng@m2-xuzheng:~$ sudo mv swap_xuzheng.* /etc/apache2/ssl/
xuzheng@m2-xuzheng:~$ sudo a2enmod ssl & sudo service apache2 restart
[1] 1712
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
[1]+  Done                  sudo a2enmod ssl
xuzheng@m2-xuzheng:~$
```

```
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

SSLCertificateFile /etc/apache2/ssl/swap_xuzheng.crt
SSLCertificateKeyFile /etc/apache2/ssl/swap_xuzheng.key
```

```
xuzheng@m2-xuzheng:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
xuzheng@m2-xuzheng:~$ sudo service apache2 reload
xuzheng@m2-xuzheng:~$ _
```

Podemos comprobar con **cURL** que m2 acepta peticiones HTTPS:

```
xzdi@DESKTOP-S58JSSA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl -k https://192.168.56.71/swap.html
<HTML>
<BODY>
SWAP M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>
```

## 3. Nginx como balanceador para peticiones HTTPS

En m3 activamos **Nginx** y aseguramos de que esté en funcionamiento:

```
xuzheng@m3-xuzheng:~$ sudo systemctl enable nginx.service
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install
Executing: /lib/systemd/systemd-sysv-install enable nginx
xuzheng@m3-xuzheng:~$ sudo systemctl start nginx.service
xuzheng@m3-xuzheng:~$ sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-04-22 17:04:25 UTC; 4s ago
     Docs: man:nginx(8)
   Process: 13400 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 13388 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 13400 (nginx)
    Tasks: 2 (limit: 4653)
   CGroup: /system.slice/nginx.service
           └─13400 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─13406 nginx: worker process

abr 22 17:04:25 m3-xuzheng systemd[1]: Starting A high performance web server and a reverse proxy server: nginx.service.
abr 22 17:04:25 m3-xuzheng systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid PID '0'.
abr 22 17:04:25 m3-xuzheng systemd[1]: Started A high performance web server and a reverse proxy server: nginx.service.
lines 1-15/15 (END)
```

Creamos un nuevo directorio *ssl* para guardar los archivos:

```
xuzheng@m3-xuzheng:~$ sudo mkdir ssl
xuzheng@m3-xuzheng:~$ sudo mv swap_xuzheng.* ssl/
xuzheng@m3-xuzheng:~$ _
```

Editamos el archivo `/etc/nginx/conf.d/default.conf` para añadir un nuevo bloque **server** con los siguientes datos:

```
location /
{
    proxy_pass http://balanceo_xuzheng;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Connection "";
}

server{
    listen 443 ssl;
    ssl on;
    ssl_certificate /home/xuzheng/ssl/swap_xuzheng.crt;
    ssl_certificate_key /home/xuzheng/ssl/swap_xuzheng.key;
    server_name balanceador_xuzheng;

    access_log /var/log/nginx/balanceador_xuzheng.access.log;
    error_log /var/log/nginx/balanceador_xuzheng.error.log;
    root /var/www;

    location /
    {
        proxy_pass http://balanceo_xuzheng;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

Reiniciamos **Nginx** y comprobamos desde el anfitrión:

```
txdl6@DESKTOP-S58Q8SA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl -k https://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

txdl6@DESKTOP-S58Q8SA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl -k https://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>
```

#### 3.1. Opciones avanzadas

A partir de la configuración anterior podemos añadir algunas mejoras. Puesto que las operaciones SSL consume recursos extras del CPU, existen dos maneras de minimizar el número de estas operaciones por cliente: la primera es habilitar la opción **keepalive** para poder mandar varias peticiones por conexión y la segunda es mediante la reutilización de parámetros de sesión SSL para evitar **ssl handshakes**. Dichas sesiones se almacenan en un cache que se configura con la directiva **ssl\_session\_cache**. El timeout por defecto de dicho caché es de 5 minutos, para modificarlo podemos utilizar la directiva **ssl\_session\_timeout**.

En el siguiente ejemplo habilitamos un cache de tamaño 1MB con un timeout de 10 minutos:

```
server{
    listen 443 ssl;
    ssl on;
    ssl_certificate      /home/xuzheng/ssl/swap_xuzheng.crt;
    ssl_certificate_key  /home/xuzheng/ssl/swap_xuzheng.key;
    server_name balanceador_xuzheng;

    ssl_session_cache    shared:SSL:1M;
    ssl_session_timeout  10m;

    access_log /var/log/nginx/balanceador_xuzheng.access.log;
    error_log  /var/log/nginx/balanceador_xuzheng.error.log;
    root /var/www;

    location /
    {
        proxy_pass http://balanceo_xuzheng;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

## 4. IPTABLES

Comenzamos esta sección comprobando el estado del cortafuegos en m1:

```
xuzheng@m1-xuzheng:~$ sudo iptables -L -n -v
Chain INPUT (policy ACCEPT 592 packets, 188K bytes)
 pkts bytes target     prot opt in     out     source                   destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source                   destination
Chain OUTPUT (policy ACCEPT 29 packets, 6397 bytes)
 pkts bytes target     prot opt in     out     source                   destination
xuzheng@m1-xuzheng:~$ _
```

Para facilitar la configuración de todas las máquinas, creamos un script en m1 con los órdenes al que posteriormente copiaremos en las demás máquinas. Para una configuración con el que sólo se permita el tráfico HTTP y HTTPS creamos el siguiente script *iptables.sh*:

```
#!/bin/sh
#Parar el cortafuegos y eliminar todas sus reglas
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X

#Denegar cualquier tráfico
iptables -F INPUT DROP
iptables -F OUTPUT DROP
iptables -F FORWARD DROP

#Permitir conexiones
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

#Permitir acceso a localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

#Permitir el tráfico HTTP y HTTPS
iptables -A INPUT -p tcp -m multiport --ports 80,443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --ports 80,443 -m state --state NEW,ESTABLISHED -j ACCEPT
```

Ejecutamos el script con **sudo bash iptables.sh** y comprobamos desde el anfitrión:

```

txdl6@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl http://192.168.56.70/swap.html
<HTML>
<BODY>
SWAP_M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

txdl6@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl -k https://192.168.56.70/swap.html
<HTML>
<BODY>
SWAP_M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

```

Sin embargo si intentamos hacer ping a m1:

```

C:\Users\txdl6>ping 192.168.56.70

Pinging 192.168.56.70 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.56.70:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

podemos ver que el tráfico no está permitido. Para configurar el cortafuegos en m2 y en m3, copiamos el script con **scp** y lo ejecutamos con **sudo bash**.

#### 4.1. Opciones avanzadas

A partir del script anterior, podemos añadir órdenes para permitir SSH (puerto 2222 en m1 y m2 y 22 en m3) , PING y DNS:

```

#Permitir SSH
iptables -A INPUT -m state --state NEW -p tcp --dport 2222 -j ACCEPT
iptables -A OUTPUT -m state --state NEW -p tcp --sport 2222 -j ACCEPT

#Permitir PING
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT

#Permitir DNS
iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT
iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
iptables -A OUTPUT -m state --state NEW -p tcp --sport 53 -j ACCEPT
iptables -A OUTPUT -m state --state NEW -p udp --sport 53 -j ACCEPT

```

Si intentamos hacer ping ahora a m1:

```

C:\Users\txdl6>ping 192.168.56.70

Pinging 192.168.56.70 with 32 bytes of data:
Reply from 192.168.56.70: bytes=32 time<1ms TTL=64
Reply from 192.168.56.70: bytes=32 time<1ms TTL=64
Reply from 192.168.56.70: bytes=32 time<1ms TTL=64
Reply from 192.168.56.70: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.70:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

podemos ver que ésta vez se consigue sin problemas. Para que sólo se acepten peticiones HTTP y HTTPS provenientes de m3 en m1 y en m2, modificamos el apartado de HTTP y HTTPS en los scripts de m1 y m2 de la siguiente forma:



```
#Permitir el tráfico HTTP y HTTPS
iptables -A INPUT -p tcp -m multiport --dports 80,443 -m state --state NEW,ESTABLISHED -s 192.168.56.72 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --sports 80,443 -m state --state NEW,ESTABLISHED -d 192.168.56.72 -j ACCEPT
```

Comprobamos desde el anfitrión:

```
xzdh@DESKTOP-S58Q8SA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl http://192.168.56.70/swap.html
curl: (28) Failed to connect to 192.168.56.70 port 80 after 21000 ms: Timed out

xzdh@DESKTOP-S58Q8SA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl http://192.168.56.71/swap.html
curl: (28) Failed to connect to 192.168.56.71 port 80 after 21003 ms: Timed out

xzdh@DESKTOP-S58Q8SA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xzdh@DESKTOP-S58Q8SA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl -k https://192.168.56.70/swap.html
curl: (28) Failed to connect to 192.168.56.70 port 443 after 21000 ms: Timed out

xzdh@DESKTOP-S58Q8SA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl -k https://192.168.56.71/swap.html
curl: (28) Failed to connect to 192.168.56.71 port 443 after 21001 ms: Timed out

xzdh@DESKTOP-S58Q8SA MINGW64 /e/DGIIIM/QUINTO 2º CUAT/SWAP/Prácticas/P4 (main)
$ curl -k https://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>
```

## 4.2. Hacer persistente reglas IPTABLES

Para que las reglas se mantengan tras el reinicio de las máquinas, necesitamos instalar en primer lugar la herramienta **iptables-persistent**:

```
xuzheng@m1-xuzheng:~$ sudo apt install iptables-persistent
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 netfilter-persistent
Se instalarán los siguientes paquetes NUEVOS:
 iptables-persistent netfilter-persistent
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 36 no actualizados.
Se necesita descargar 13,8 kB de archivos.
Se utilizarán 89,1 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Al responder sí para continuar nos muestra la siguiente vista:

```
Configuración de paquetes

Configuración de iptables-persistent

Las reglas actuales de iptables se pueden guardar en el archivo de configuración
/etc/iptables/rules.v4». Estas reglas se cargarán automáticamente durante el inicio del
sistema.

Las reglas sólo se guardan automáticamente durante la instalación del paquete. Puede
consultar las instrucciones para mantener el archivo de reglas actualizado en la página de
manual de «iptables-save(8)».

¿Desea guardar las reglas de IPv4 actuales?
<Yes> <No>
```

El cuál nos pregunta si queremos guardar la configuración actual. Le damos que sí y nos aparecerá otra vista similar para ip6, respondemos que sí de nuevo y se finalizará la instalación. Si en algún momento volvemos a modificar el cortafuegos podemos actualizar la configuración persistente utilizando **sudo bash -c “iptables-save >/etc/iptables/rules.v4”** y **sudo bash -c “ip6tables-save >/etc/iptables/rules.v6”**.

## 5. Bibliografía

- <https://www.openssl.org/docs/manmaster/man1/openssl-req.html>
- [http://nginx.org/en/docs/http/configuring\\_https\\_servers.html](http://nginx.org/en/docs/http/configuring_https_servers.html)
- [http://nginx.org/en/docs/http/nginx\\_http\\_ssl\\_module.html#ssl\\_session\\_cache](http://nginx.org/en/docs/http/nginx_http_ssl_module.html#ssl_session_cache)
- <https://linux.die.net/man/8/iptables>
- <https://www.layerstack.com/resources/tutorials/How-to-enable-and-disable-Ping-from-IPTables-on-Linux-Cloud-Servers>
- <https://linuxconfig.org/how-to-make-iptables-rules-persistent-after-reboot-on-linux>
- <https://askubuntu.com/questions/230476/how-to-solve-permission-denied-when-using-sudo-with-redirection-in-bash>